Intranet Security technologies - Sesame or SSL?

Paul Ashley[†]
Gary Gaskell[†]

[†] Information Security Research Centre
Queensland University of Technology
GPO Box 2434 Brisbane 4001 Australia

http://www.isrc.qut.edu.au
{ashley,gaskell}@isrc.qut.edu.au

Joris Claessens[‡]
Mark Vandenwauver[‡]

[‡] Computer Security and Industrial Cryptography
Katholieke Universiteit Leuven
ESAT/COSIC Kardinaal Mercierlaan
94 3001 Heverlee Belgium
http://www.esat.kuleuven.ac.be/cosic
{Mark.vandenwauver,joris.claessens}@esat.kuleuven.ac.be

Abstract

In recent years there has been a phenomenal increase in the use of networking. The technologies of the Internet are becoming the standard for organisational information retrieval. Due to their success it has become possible to create a virtual office environment for an organisation, regardless of the geographical diversity of the personnel. As such the security of an organisation's information is in doubt unless steps are taken to protect the information assets.

Cryptographic solutions are essential to protect computer assets where the solutions are deployed over untrusted networks. With the value of information on an organisation's network and the complexity of the IT infrastructure it is often inadvisable to place a high degree of trust even on internal networks. The internal networks are likely to have components that are interconnected using infrastructure that is outside of the control of the owner of the information. Hence the use of cryptographic techniques is required to enhance the security of an organisation's information as it travels and is accessed over untrusted networks.

Generically speaking the threats to the information resources include integrity of the information and loss of confidentiality (access by unauthorised persons). This paper compares SSL and SESAME by discussing: what security services they provide, what cryptographic technology is used by each solution, the applications supported, where they fit in the TCP/IP model, availability and finally a description of their limitations. The objective is to compare these security technologies with a view to using them for Intranet security solutions.

This paper concludes that SSL is a strong security solution for some Internet applications, however essential services such as authorisation are missing from SSL and an application level architecture such as SESAME is highly desirable. It also concludes that SESAME provides a level of services well above that of SSL and thereby greatly assists an organisation in the efficient management of its IT security.

1 Introduction

An Intranet is becoming a core component of most organisations infrastructure. The Internet technologies used in an Intranet are seen as an effective way to exchange information. Future growth in the use of Intranets and the Internet depends very heavily on how well they are secured. The problem of securing networks can be approached in numerous ways, the most popular ones to date are:

Appeared in *Proceedings of the AUUG'98 Conference*
Sydney, Australia, September 16-18, 1998, pp 133–142

- Application level security (for example: corporate applications, telnet, ftp, ...)
- Networking level security. This might be achieved by adding security at various layers in the network hierarchy (for example networking, transport...)
- Physical security

There are likely to be many different security needs. It is important to gain an understanding of what security services and features are required. Some examples may be: authentication of users and network entities, data protection in transit (integrity, confidentiality and authentication of sender), non-repudiation of data receipt, access control for users, single sign-on solutions, key distribution, auditing and the security of system bootstrapping. There are also other considerations when designing a network solution such as cryptographic policy and key recovery requirements.

The purpose of this paper is to examine two recent solutions for network security, SESAME and SSL, and their suitability for Intranet security solutions. The paper aims to highlight the differences in the two solutions. The paper is structured as follows. The first section gives a brief overview of the SESAME and SSL technologies. The next section compares these two solutions. From this comparison we conclude that SSL may not be suitable for a majority of Intranet deployments due to its lack of user authorisation services and due to the superior manageability of security by SESAME.

The paper presents the supporting arguments for this conclusion and finishes with some recommendations.

2 An Overview of the SESAME and SSL Technologies

2.1 SESAME

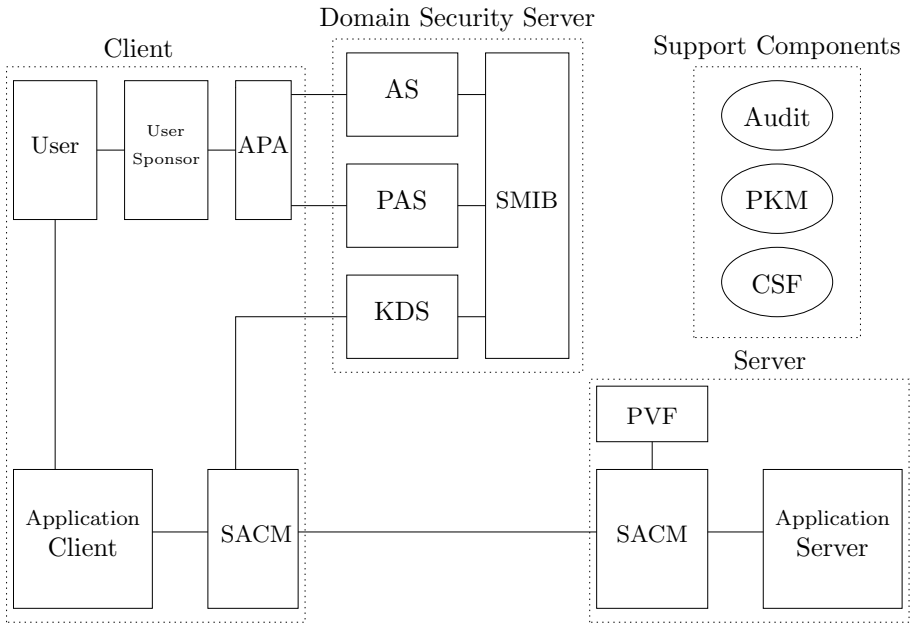

Figure 1: Overview of the SESAME components

Appeared in *Proceedings of the AUUG'98 Conference*
Sydney, Australia, September 16-18, 1998, pp 133–142

SESAME [10, 11, 15] is the name of a security architecture. As a security architecture it describes where in a system various security services are required.

It is the result of a collaboration of Bull, ICL and Siemens together with some leading European research groups. The project was funded by the EC under the auspices of its RACE program. SESAME is an acronym for *"A Secure European System for Applications in a Multi-vendor Environment"*. Figure 1 gives an overview of the SESAME architecture. At first glance it looks very complex but it is possible to distinguish four boundaries in the architecture: the client, the domain security server, the (application) server, and the support components. The complexity corresponds to the level of services provided.

The client system incorporates the User, User Sponsor (US), Authentication Privilege Attribute (APA) Client, Secure Association Context Manager (SACM) and client application code. The User Sponsor is the user's interface to the SESAME system. This allows the user to logon. The APA is used by the User Sponsor for the communication with the domain security server. The SACM provides the data protection services (data authentication, data confidentiality, non-repudiation) for the client-server interaction.

The Domain Security Server is very similar to the Kerberos one [12]. The main difference is the presence of the Privilege Attribute Server (PAS) in SESAME. The server has been added to manage the access control mechanism that is implemented by SESAME. Because of its many advantages SESAME has opted to implement role based access control (RBAC) [16]. This scheme is enforced using Privilege Attribute Certificates (PACs) [6]. The function of the Authentication Server (AS) and Key Distribution Server (KDS) (ticket granting server in Kerberos) are similar to their Kerberos counterparts: providing a single sign-on and managing the cryptographic keys. A major difference with Kerberos is that SESAME also supports public-key based authentication using the X.509 authentication mechanism [9].

When the application server receives a message from an application client indicating that it wants to set up a secure connection, it forwards the client's credentials and keying material (an encrypted session key) to the PAC Validation Facility (PVF), which checks whether the client has access to the application. If this check is successful, it decrypts the keying material and forwards the session keys (SESAME uses independent keys for providing data authentication and data confidentiality) to the SACM on the server machine. Through this, the application server authenticates to the client (mutual authentication) and it also enables the application server to secure the communication with the client.

The SESAME architecture provides a number of support components used throughout the system. These include the Audit facility (providing detailed audit logs), Cryptographic Support Facility (CSF) (providing the various cryptographic primitives), and the Public Key Management (PKM) facility.

2.2 SSL

The most dominating factor in the growth of the Internet in recent years has been the success of the World Wide Web (WWW). The *Secure Sockets Layer (SSL)* solution [7] from Netscape Communications is an attempt to provide security for the WWW. It is also used to secure other applications such as telnet and ftp.

The original version of SSL was SSL 2.0 [8]. This version contained a number of security flaws. These flaws were solved in the SSL 3.0 specification [7]. Both versions 2.0 and 3.0 are currently used and are a de facto standard. In 1997 the IETF formed a *Transport Layer Security (TLS)* working group which has adopted the SSL 3.0 protocol in its initial release of their TLS protocol [5] (including some minor changes). This protocol is intended as a generic solution to secure the transport layer (see Figure 3) and would be independent of the actual application.

In the remainder of this paper, SSL is used to denote the SSL/TLS standard.

SSL is situated underneath the application layer. It provides entity authentication, data authentication and data confidentiality, where the entity authenticated is the workstation. The structure of the SSL protocol is outlined in Figure 2. It consists of two levels: the Record Layer Protocol and four other protocols of which the most important is the Handshake Protocol.

At the beginning of a session, the client and server perform a handshake, during which both entities negotiate on the algorithms and cryptographic keys that are going to be used. Both entities (i.e. the client and server machines) are also optionally authenticated. Strictly speaking it is the machines that are authenticated and not the users. The SSL specification does not define 'client authentication'. What exactly client means is open to interpretation, however at the sockets layer we take this to be the machine. The new security parameters are used after the 'changecipherspec' message at the end of the handshake.

All data is sent through the Record Layer which provides three functions: fragmentation, compression and cryptographic protection. No compression algorithms are defined yet. Cryptographic protection means including a MAC for data authentication and using encryption for data confidentiality.

Client and server do not always have to perform a complete handshake. It is possible to use the same security parameters negotiated in the previous connection. In SSL terminology, this means that a new connection can be established within the same session.

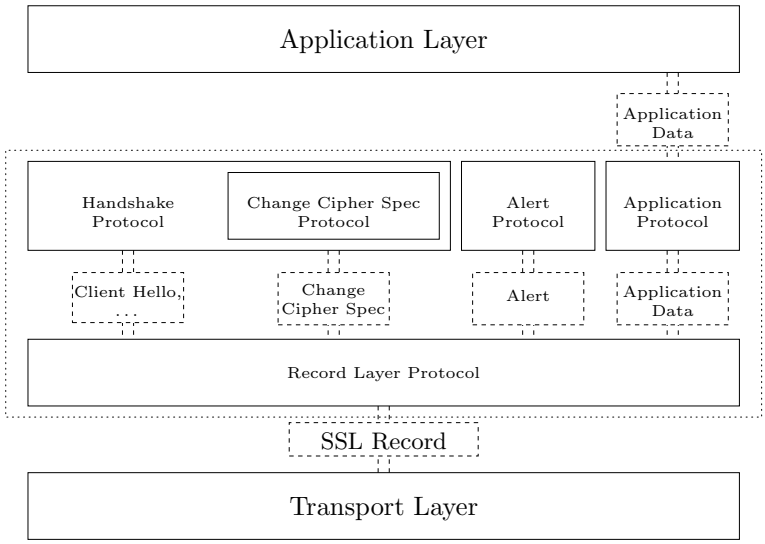

Figure 2: SSL Structure

3 Comparison

This section attempts to describe the differences between SESAME and SSL.

3.1 Positioning in the Networking Model

Figure 3 shows the OSI and TCP/IP reference models and the relationship between them. The OSI model [14] has seven layers and is shown on the left of the figure. The TCP/IP model [4] has four layers and is shown on the right. The figure also shows where SESAME and SSL are

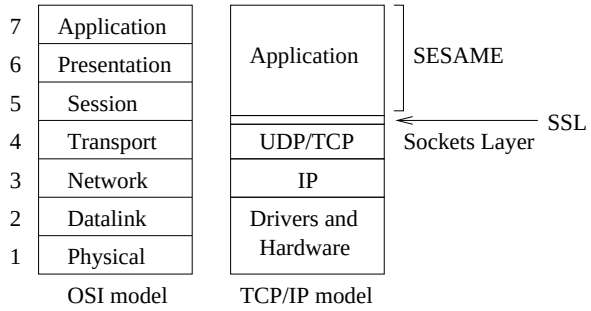

Figure 3: The Positioning of SESAME and SSL in the TCP/IP Model

situated in the model. SESAME is an application level solution, whereas SSL is a transport layer solution (or more specifically a socket layer solution). This socket layer has recently been proposed as a layer that could be added to the TCP/IP model between the transport layer and application layer.

Placing the security solutions at alternate layers in the model causes a number of differences:

- How the applications interact with the security services;
- What security services are available to applications.

Applications that use security services at the application layer have to be modified to use these services. In the case of SESAME, applications must be modified to make calls to the SESAME security libraries, often using the Generic Security Services API (GSS-API). This means that the application code not only performs the function of the application, but must also be aware of the security services it would like to use. This also has advantages, as the application and/or PAC Validation Facility (PVF) is aware of the user that wishes to take an action. Often it is the application layer only that knows whether this action is valid or otherwise. Providing security services at the socket layer means that applications (at least in theory) do not have to be as security aware (in some cases not at all). In the case of SSL, applications that are built using the normal socket library, can be SSL secured by simply rebuilding them with a new SSL socket library. This means it is possible to turn an insecure application into a secure one without changing a line of application code.

The difference in the security services that can be provided at the different layers is described in the next section. In general the higher the security solution is in the hierarchy, the greater the possibility is to provide additional services.

3.2 Security Services Provided

Table 1 gives an overview of the different security services that are provided by SESAME and SSL. The table indicates that the services that SSL provides are in fact a subset of the services provided by SESAME. This is mainly due to the fact that SESAME is situated in a higher layer of the model. It is however also important to note that SESAME is a whole architecture, while SSL is just a standard that defines in what way the communication between two parties should be secured. The protocol is restricted to a description of the algorithms to use and the format and content of the messages.

The first security service that is only offered by SESAME is a single sign-on for users. That is, SESAME allows users of the SESAME secured network to log on once to the network, be

Table 1: Security services

Security Service	SESAME	SSL
Single Sign-on for Users	yes	(no)
User authentication	yes	(yes/no)
Workstation authentication	(no)	yes
Data confidentiality	yes	yes
Data authentication	yes	(yes)
Key distribution	yes	yes
Non-repudiation	yes	(no)
Auditing	yes	(no)
Access control	yes	(no)

provided with SESAME credentials, and then to use these to access resources across the network. The advantages of a single sign-on solution have been recognised for some time with Kerberos being the most well known example. A single sign-on service is not defined in SSL, as SSL is really a peer to peer protocol, however an application that uses SSL can implement some kind of single sign-on mechanism. e.g., browsers only prompt once for a password, from the moment the user has provided this password, sessions can be established without extra input of the user (providing that they are connecting to the same WWW server). The security of such techniques are often in question, due to the way browsers protect the secrets that have been given up to them.

In both technologies data is protected while in transit with options for both confidentiality and integrity protection. Both solutions provide facilities for key distribution, e.g., session key establishment. SESAME provides a helpful extension by allowing for a site-configurable option of different strength confidentiality and integrity mechanisms.

Another defining difference depends on your interpretation of the SSL specification. The SSL specification discusses 'client authentication' however, it does not define what the client is. The client is commonly the machine rather than the real user of the machine. The issue is further complicated by the position of the user in the application layer and the machine in the sockets layer. Some applications have required the user to present a passphrase, before allowing the SSL implementation on the machine access to the private key. If the private key is not protected by a passphrase (as in some SSL telnet and ftp implementations) then we contend the user is not authenticated by the use of the public key cryptography in SSL. In this situation it is only the machine that is authenticated. Further, we note that it is common for an application to use SSL to secure the channel of a communications session and to use a username and password tuple in the application layer to authenticate the user. We admit that some will disagree with this position. However, we submit that SESAME presents genuine user authentication, as it is at the application layer, and that the use of SSL for user authentication is debatable.

SSL operates at the sockets layer and is therefore oblivious to to the actions of a user. Hence, at least from a legal viewpoint, it cannot provide a non-repudiation service. SESAME on the other hand provides services directly to the application and hence the non-repudiation of user actions is possible. It can be argued that non-repudiation is not a widely used service, at least in commercial applications. However this is expected to change on two fronts. One obvious driver is electronic commerce. The other may be the auditing and management requirements in large world-wide intranets (that are now deployed by multi-national corporations).

An extensible set of audit tools is provided by SESAME. SSL does not define a method of auditing. However, a lot of products implement a separate logging mechanism. For example

secure web servers can log specific SSL related interactions. On the other hand, protection of these log files is as important as creating them, and this is certainly not always guaranteed.

One of the main features of SESAME is access control, and in particular Role Based Access Control (RBAC). SSL does not implement an access control service. To provide some kind of access control to web servers it is possible to use the username/password technique (protected by SSL) or the client authentication (using its private key) provided by SSL.

There is a high administrative overhead with separate passwords to protect the internal web pages of an intranet. It is these situations where the management benefits of RBAC can be realised. SESAME provides RBAC access control services and as such we believe its management of access control is an order of magnitude better than SSL.

3.3 Algorithms and technology

Table 2 gives an overview of all the cryptographic primitives and algorithms that are used in the different systems.

In the SESAME column, the 'x' denotes the fact that this algorithm is used in the public release. The SESAME technology however does not depend on specific algorithms. Due to the modular structure, it is normally easy to add or replace other algorithms.

In the SSL column, the 'x' means that this algorithm is defined in the standard, in other words, that an identifier of this algorithm has been agreed upon. It is however not necessary to implement every defined algorithm to have an SSL compliant application.

Table 2: Cryptographic algorithms and technology used

Algorithm		SESAME	SSL
Certificates	X.509v3	x	x
Key distribution/agreement	Kerberos	x	
	RSA	x	x
	Diffie Hellman	(x)	x
	Fortezza	(x)	x
Bulk encryption	DES-CBC	x	x
	DES-EDE3-CBC	(x)	x
	IDEA-CBC	(x)	x
	RC2-CBC	(x)	x
	CDMF-CBC	(x)	x
	RC4	(x)	x
	Skipjack	(x)	x
Hash	MD5	x	x
	SHA-1	(x)	x
MAC	HMAC	(x)	x
	DES-CBC-MAC	x	
	RIPEMD-160	(x)	
Digital Signatures	RSA	x	x
	DSS	(x)	x
Protection methods	Timestamps	x	
	Sequence numbers	x	x
	Nonces	x	x
Miscellaneous	Smartcards	x	(x)

Table 3: Implementations and Supported Applications

Package	Description	Where
SSLeay	freeware library, source code	http://www.ssleay.org
SSLRef	non-exportable reference library	http://home.netscape.com
SSLPlus	non-exportable commercial library	http://www.consensus.com
SSLava	Java library	http://www.phaos.com
iSaSiLk	Java library	http://jcewww.iaik.tu-graz.ac.at
SSLapps	SSLeay based applications (TELNET, FTP, Apache, Mozilla)	http://www.ssleay.org
Commercial	mostly export restrictions	Netscape, Microsoft, ...
SESAME lib.	free to use under license conditions, full source code	[15]
Applications	TELNET, FTP, RPC, ...	[2]
Commercial	security servers, GSS-API	http://www.ism.bull.net, http://www.icl.co.uk
Commercial	Intranet WWW system Siemens	http://www.trustedweb.com

3.4 Applications and Availability

Table 3 is intended to be an overview of the different available implementations of SSL and SESAME. It shows both the libraries and supported applications. It is not an exhaustive list.

At this stage SSL has been focused on providing security for WWW transactions. Also telnet and ftp have been secured by SSL. SESAME has been used to secure Intranet applications like the rtools, RPC (Remote Procedure Call), NFS (Network Filesystem) and also telnet and ftp. There exists also a sesamized WWW application for use in an Intranet. This is available as a commercial product.

SSL is particularly designed for and is suitable for WWW interactions. The security services required for these type of transactions are authentication of client and server, session key negotiation and data protection in transit. SSL has been designed specifically for this purpose and provides more options and better performance than SESAME in this simple configuration. The WWW, in its current form does not require other services such as a finer grained access control (although this is considered a weakness by some as fine grained access control is deemed essential for future WWW development and is indeed required in the intranet environment). SSL is suitable for telnet and ftp for similar reasons as the current model of the WWW.

SESAME is more suitable for applications where finer grained access control is required and this is shown in its ability to be able to secure the Network Filesystem. The implementation allows users to be given privileges at login, and the system controls access to files using these SESAME privileges.

SESAME provides security for applications by providing the Internet Standard Generic Security Service Application Program Interface (GSS-API) [13]. As previously described, the application code has to be modified to call these routines in the appropriate places. Experience has shown that the GSS-API is an excellent tool for securing applications [3].

It should also be stated that due to its high level of services, SESAME is often integrated into an organisation's custom software.

3.5 Legal regulations and restrictions

When discussing implementations of cryptography, it is unfortunately often necessary to mention legal and cryptographic policy issues. Some countries (U.S., France, U.K. and Australia) have a specific policy regarding import and, especially, export of cryptographically enhanced products.

The U.S. export restrictions have serious implications on the level of security that can be obtained with SSL enhanced products, in particular the most popular WWW browsers and servers. Until January 1997, the strongest cryptography that U.S. companies were allowed to export was limited to a 40-bit security level. Recently, the export options have been increased to a 56-bit level, although certain restrictions still apply.

A lot of creative ways are possible to achieve strong cryptography, like tunneling and proxy mechanisms. In October'97 McKay [**?**] published a program with which the Netscape browser could be patched to a version with strong crypto support, just by changing a few bytes of the executable. In April'98, Netscape released the source code of its Communicator browser. Immediately following the release, the Mozilla Crypto Group started with the integration of the SSLeay library in the browser, which gives it full strength cryptography.

Also SESAME is affected by legal regulations. The SESAME project was forced to replace the DES algorithm in the SESAME distribution by a simple XOR operation. Therefore anyone who installs the SESAME software should replace this by a real DES implementation. Because of the nice approach of pluggable crypto, this is a fairly easy task and can be done by using a separate CSF (see section 2.1). In fact, this has already been done for the Linux version which can be downloaded from [2].

3.6 Other limitations

Both SESAME and SSL share a common limitation: they both use only session oriented protocols (they insist on a session being established before security services can be provided). In numerous applications this type of protocol is suitable, examples are WWW transactions, telnet, ftp, BSD rtools, remote procedure call and NFS. However certain applications do not require and cannot use session oriented protocols. For example email security relies on sessionless protocols (you certainly want to be able to receive a secured email without first establishing a session). It would certainly be advantageous to both SESAME and SSL if they also included support to secure sessionless protocols. The IETF has drafted the Independant Data Unit Protection (IDUP) IETF Internet Draft [1]. IDUP attempts to extend the GSS-API to support security for a generic data unit that is not part of a security context.

Adding security to an application always causes a decrease in performance. This decrease is not only due to the cryptographic computations, but also to the extra data that has to be exchanged, especially for session establishment. SESAME requires large tokens (2000 bytes) to be sent for establishing a session. This is comparable to the total length of the handshake messages in SSL. However, in reality this does not form a problem.

4 Conclusion

SESAME and SSL have been designed for different purposes. SESAME is a comprehensive solution for providing single sign-on, and a range of security services for Intranet applications. The security services are provided at the application layer. SSL is designed for providing secure connections to Internet applications by placing its security at the transport layer. It provides numerous choices for cryptographic solutions.

SESAME and SSL could be considered complementary technologies. However SESAME provides more services and is a more manageable solution.

References

[1] C. Adams. Independant data unit protection generic security service application programming interface (idup-gss-api). IETF Internet Draft, May 1998. ftp://mirror.aarnet.edu.au/ietf/internet-drafts/draft-ietf-cat-idup-gss-11.txt.

[2] P. Ashley. ISRC SESAME Application Development Pages, http://www.fit.qut.edu.au/~ashley/sesame.html.

[3] P. Ashley, M. Vandenwauver, and B. Broom. A Uniform Approach To Securing Unix Applications Using SESAME. In *Information Security and Privacy - Proceedings of the Third Australian Conference on Information Security and Privacy ACISP'98*, pages 24–35, Brisbane, Australia., July 1998. Lecture Notes in Computer Science - Springer Verlag.

[4] S. Bellovin. Security Problems in the TCP/IP Protocol Suite. *Computer Communications Review*, 19(2):32–48, April 1989.

[5] T. Dierks and C. Allen. The TLS Protocol Version 1.0, November 1997. Internet Draft.

[6] ECMA 219. ECMA-219 Security in Open Systems - Authentication and Privilege Attribute Security Application with Related Key Distribution Functionality, 2nd Edition, March 1996. European Computer Manufacturers Association.

[7] A.O. Freier, P. Karlton, and P.C. Kocher. The SSL Protocol Version 3.0, March 1996. Internet Draft.

[8] K. Hickman and T. Elgamal. The SSL Protocol, 1995. Internet Draft.

[9] ITU. ITU-T Rec. X.509 (revised). The Directory - Authentication Framework, 1993. International Telecommunication Union, Geneva, Switzerland.

[10] P. Kaijser. SESAME The European Solution to Security For Open Systems. In *Proceedings of the 10th World Conference on Computer Security, Audit and Control COMPSEC*, pages 289–297, London, UK, October 1993.

[11] P. Kaijser, T. Parker, and D. Pinkas. SESAME: The Solution To Security for Open Distributed Systems. *Computer Communications*, 17(7):501–518, July 1994.

[12] J. Kohl and C. Neuman. The Kerberos Network Authentication Service V5, 1993. RFC1510.

[13] J. Linn. Generic Security Service Application Program Interface Version 2, 1997. RFC2078.

[14] OSI. OSI Reference Model - Part 2 : Security Architecture. ISO Information Processing Systems, ISO, Geneva, Switzerland, 1988. ISO 7498-2.

[15] M. Vandenwauver. The SESAME home page, http://www.esat.kuleuven.ac.be/cosic/sesame.

[16] M. Vandenwauver, R. Govaerts, and J. Vandewalle. How Role Based Access Control is Implemented in SESAME. In *Proceedings of the 6-th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 293–298. IEEE Computer Society, 1997.